A DISTRIBUTED WAVELET APPROACH FOR EFFICIENT INFORMATION REPRESENTATION AND DATA GATHERING IN SENSOR WEBS

Sungwon Lee, Sundeep Pattem,
Godwin Shen, Aaron Tu,
Bhaskar Krishnamachari, Antonio Ortega

Department of Electrical Engineering
University of Southern California
Los Angeles, CA

Michael Cheng, Sam Dolinar,
Aaron Kiely, Hua Xie

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA

ABSTRACT

In this work we investigate novel approaches for information processing and representation in a sensor web. Sensor nodes capture information that is both temporally and spatially correlated. Exploiting spatial correlation requires data exchange between sensors, which should be minimized in order to keep power consumption low and maximize the life of the system. We are investigating methods for sampling (which sensors should make measurements), routing (how does the information flow towards a fusion center), processing (how to perform a wavelet transform along a network route) and compression (how to compress the output of the wavelet transform). All of these aim at maximizing the quality of the data available at the fusion center for a given energy consumption target at the nodes. In this paper we will report algorithmic, analytical and implementation progress made in this work over the last year.

1. INTRODUCTION

Wireless sensor networks (WSN) can offer mobility and versatility for a variety of applications, such as object detection/tracking, environment monitoring and traffic control [4]. Still, one of the main obstacles they face is that they often rely on batteries for power supply; thus limiting their energy consumption becomes essential to ensure network survivability.

When data is acquired at multiple correlated sources, aggregation involving in-network data compression can offer a more efficient representation of measurements, significantly reducing the amount of information that needs to be transmitted over the network, thus leading to a potentially large reduction in energy consumption. Prior work has addressed a number of distributed source coding (DSC) methods as a means to decorrelate data. While some rely on information exchange and additional computation inside the network to propose distributed versions of transforms, such as Karhunen-Loève [16] and wavelets [25], others propose schemes that do not require internode communication, such as networked Slepian-Wolf coding [11, 23]. In general, DSC techniques face a trade-off between i) more processing at each node to achieve more compression and ii) less processing which would require more information (bits) to be sent to the sink. This trade-off has also been addressed by previous research. [22] provides an analysis on the regions in a network that should favor compression over routing based on the impact of spatial correlation of the measurements. The performance of aggregation under a more general data model is considered in [18].

This work has been funded in part by the NASA Earth Science Technology Office under grant AIST-05-0081.

Our focus has been on the problem of finding an optimal assignment of compression algorithms to nodes, in the sense of minimizing the energy consumption, when different methods are available to choose from. Since the distortion/energy consumption tradeoff also depends on factors such as network topology and medium characteristics, different coding methods may be better suited for different parts of the network. These methods can consist in simple coding schemes such as DPCM, or more complex ones, such as wavelet transforms with an arbitrary number of levels of decomposition.

The basic trade-off we have investigated is in the selection of number of levels of decomposition for a wavelet transform, but the same principle can be extended to other classes of signal representation and compression. We seek to achieve efficient signal compression by exploiting spatial signal correlation (e.g., temperature measurements in neighboring nodes in a sensor network will tend to be similar). In general, coding schemes that remove correlation across multiple nodes will tend to lead to higher coding efficiency, but at the cost of increased "local" communications, i.e., a distributed approach means that nodes have to exchange data before the final compressed version (which is sent to the fusion node) can be generated.

Prior work within our team [6–9] led to the development of a distributed wavelet transform, where data is forwarded along routes towards the sink and each node contributes to data decorrelation (and more efficient compression) by performing wavelet transform operations on some of the data passing through it. In particular, our work proposed techniques (based on dynamic programming) to optimize the choice of wavelet transform for a given routing structure [8] and then demonstrated that improved overall performance could be achieved by selecting jointly routing and wavelet transform [9].

This paper describes our recent progress in extending, under funding from the NASA-ESTO AIST program, the system proposed in [9]. Section 2 provides a brief description of our original distributed wavelet transform tools. Where simplifying assumptions were made in the original system we now consider extensions that in some cases lead to very significant performance improvements. Originally, no entropy coding was used on the quantized wavelet coefficients and the same number of bits were used for all wavelet coefficients of a given type (e.g., all low pass coefficients were assigned the same number of bits). We are investigating entropy coding techniques optimized for our system (Section 3) as well as bit allocation tools (Section 4) that allow us to assign number of bits in a much more flexible manner. While in our previous work we simply considered standard 1D wavelet transforms and adapted them to our scenario, we are now considering extensions to allow 2D filtering (Section 5) and to more closely link filtering and routing design via compressed sensing (Section 6). We also consider the design of erasure-correcting codes to ensure reliable delivery in our system (Section 7). We have already started implementing various aspects of the system using programmable sensors with an eye towards testing our system both in-lab and within a small scale real-life deployment (Section 8). To conclude, we summarize the project status briefly in Section 9.

2. BASELINE SYSTEM

In [6, 7] we introduced energy-aware distributed wavelet compression algorithms for WSN [6] and introduced a partial coefficient approach based on the lifting implementation [7]. Our goal was to generate the wavelet transform coefficients at the sensors, at the expense of a little extra energy spent with a few "local" transmissions, i.e., data transmissions between neighboring nodes that are needed to actually compute the wavelet transform coefficients, since the transform operates by filtering "across nodes". If the original data has sufficient spatial correlation, after quantization the wavelet coefficients can represent the original measurements using fewer bits, and the overall energy consumption in the network is lowered by reducing the amount of information that has to be transmitted. Our proposed partial coefficient approach [7] essentially allows all wavelet transform operations to be causal, in the sense of that data is processed as it is being forwarded to the central node, so that only data from nodes already traversed is used to compute the wavelet coefficients. This requires the computation and quantization of "partial" coefficients, which are transmitted over a few hops, before being used to generate the final wavelet coefficients.

We assume that a sensor network acquires measurements from a correlated data field. We consider data aggregation (compression) along a 1-D path from an edge to the sink (Fig. 1). This path is assumed known, which implies that a routing algorithm has been applied to the network first. Each sensor is assigned a number n, starting from the edge. The network topology (internode distances) is known, and each node in the 1-D path can operate using a coding scheme chosen from a predefined set of available coding schemes. In [8, 9], available schemes are discrete wavelet transforms using the same filterbank but with different number of levels of decomposition: when the number of levels decomposition is increased, the potential compression efficiency also increases (if data is highly correlated across sensors), but at the cost of more local information exchange (because data from more nodes is needed to compute some of the wavelet coefficients).

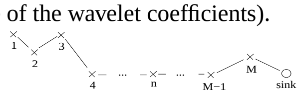

Fig. 1. 1-D path with M nodes to the sink.

Since the wavelet transform is critically sampled, the number of wavelet coefficients generated is equal to the number of nodes. Using the partial coefficient approach [7], the wavelet coefficient corresponding to node n is computed in steps: at node n a partial version of the coefficient is first generated, which becomes a full coefficient as it "incorporates" additional data from future nodes (i.e., nodes closer to the sink). The number of hops required until a partial coefficient becomes full depends on the specific transform filters being used, refer to [7] for details.

It can be shown [8] that costs can be attached to each possible coding strategy for a given node (number of bits and number of levels in the wavelet decomposition). Because of the structure of the problem, it can also be shown that costs for a given node depend only on the physical position of the node in the network and the coding scheme being used. Therefore, choosing the best coding strategy leads to solving for the minimal cost path in a state transition diagram that represents the various coding methods [8]. The optimal solution can be found using dynamic programming [10].

The gains achievable by selecting the best coding scheme are illustrated by testing an input process data that was created using a second order AR model, with poles placed such that a reasonably smooth output would be generated from white noise (poles were at $0.99e^{\pm j\frac{\pi}{64}}$). Figure 2 shows the energy consumption of different single-scheme methods (only one coding scheme for the whole network) at different distortion levels, in a network with 3 clusters of 5 sensors each (internode distance of 2m, intercluster distance of 37m).

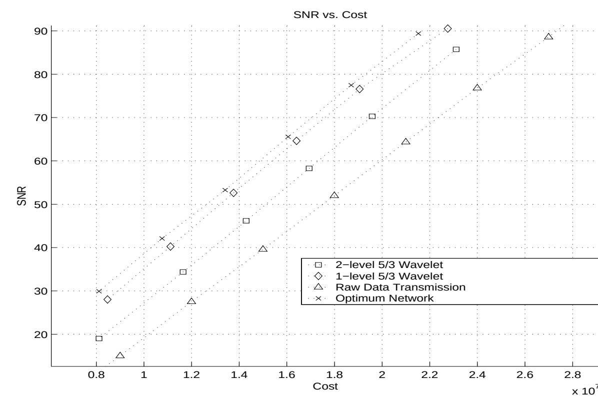

Fig. 2. Energy consumption comparison between methods with 3 clusters of 5 sensors each.

For this network, the optimum configuration such that energy consumption is minimized, obtained by the proposed dynamic programming framework is shown in Fig. 3.

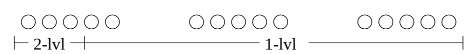

Fig. 3. Optimum network configuration obtained for simulation in Fig. 2.

Although the results suggest changes for just a few sensors when compared to the best single-scheme method, in general, such a behavior cannot be predicted beforehand. Network performance can be affected by a number of factors like the coding schemes being used, network topology, number of sensors, medium properties, data correlation, just to cite a few. Thus, a single-scheme approach might not necessarily result in the near-optimal performance. Optimization still proves to be necessary to point out the configuration that will lead to the lowest energy cost. For the simulated case shown in Figure 2, for same distortion levels, the optimum network consumed around 6% less energy than the best single-scheme method of Figure 2 (1 level wavelet) and around 32% less energy than simple raw (quantized) data transmission.

3. ENTROPY CODING

Our previous work did not explicitly consider variable length encoding of the outputs of the distributed wavelet transform. Here we address the task of using entropy coding to minimize the communication cost between sensor nodes. To simplify our analysis, we assume unidirectional transmission in a sequence of equally spaced nodes with no path merges. In such a network, Figure 4 illustrates the information that must be communicated to compute a distributed wavelet transform across nodes. Using a single stage 5/3 integer wavelet transform, a pair of DWT coefficients $\{y[2n], y[2n+1]\}$ are calculated as:

$$
\begin{aligned}
y[2n+1] &= x[2n+1] - \lfloor \frac{1}{2}(x[2n+2] + x[2n]) \rfloor \\
y[2n] &= x[2n] + \lfloor \frac{1}{4}(y[2n+1] + y[2n-1]) + \frac{1}{2} \rfloor .
\end{aligned}
$$

Here $y[2n+1]$ is the high-pass DWT coefficient, $y[2n]$ is the low-pass DWT coefficient, and $x[i]$ is the sensor data input at node i. Because communication is unidirectional (forward only), the values of $y[2n]$ and $y[2n+1]$ cannot be computed until node $2n+2$.

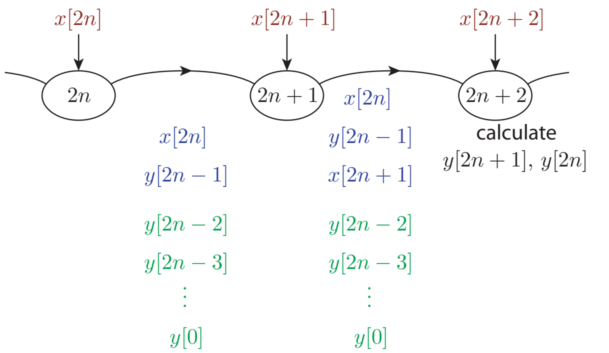

Fig. 4. Information that must be communicated in a distributed wavelet transform framework. The update quantities $\{x[2n], x[2n+1], y[2n-1]\}$ (in blue) are needed at node $2n+2$ to calculate the new DWT coefficient pair; the history quantities $\{y[0], y[1], \cdots, y[2n-2]\}$ (in green) are the previously calculated DWT coefficients that must be relayed to the sink.

Information to be transmitted falls into two categories: *update* information needed to compute the new DWT coefficient pair, and *history* information communicating the values of DWT coefficients computed at past nodes. Our entropy coding task is to efficiently encode a quantized version of the history and update information. We assume that update quantities are encoded at higher fidelity (i.e., using a quantizer with a smaller stepsize) [6,7]. The issue of selecting quantization stepsizes (bit allocation) is addressed in Section 4.

At each even-indexed sensor node $2n$, the DWT coefficient pair $\{y[2n-1], y[2n-2]\}$ is calculated using the update quantities. The high-pass coefficient $y[2n-1]$ is quantized and encoded directly; to encode $\tilde{y}[2n-2]$, the quantized version of $y[2n-2]$, the difference $\tilde{y}[2n-2] - \tilde{y}[2n-4]$ is computed and encoded to exploit some of the remaining correlation. We assume that the cost of computation at each node is much less than the cost of communication. With this in mind, each node can perform the inverse DWT using the quantized history information to produce estimates of past sensor sample values $\hat{x}[i]$. To encode the sensor input data $x[2n]$, the difference between $x[2n]$ and $\hat{x}[2n-2]$ is computed and encoded using a variable length code.

For our variable length codes we use the family of Golomb codes [19] [20]. Golomb codes are known to be optimal for geometric distributions of nonnegative integers [14]. An important step in coding is to determine the value of the code parameter m to minimize the average code length for a given distribution. We adopt the sequential parameter estimation method used in LOCO-I [31] image compression. In this method the parameter m is chosen to be the smallest power of 2 that is greater than the average absolute value of past observed sequence. In our framework, these values are readily available as we decode the history information from past nodes.

As an example, Figure 5 shows the expected transmission cost at each node using this entropy coding strategy neglecting the overhead cost of "learning" data statistics at earlier nodes. We can see that the transmission cost for DWT history quantities dominates the total cost at each node. Therefore it is more important to effectively encode history quantities than update quantities. For many sensor web scenarios the most relevant objective may be to minimize the maximum transmission cost at any node, which, in our model, is the last node in the chain.

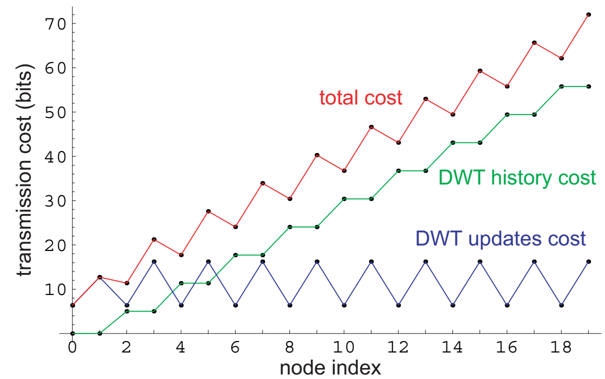

Fig. 5. Example of transmission cost at each node using entropy coding in a distributed wavelet transform framework.

Figure 6 shows the rate distortion performance achieved by (a) the proposed entropy coding approach in a distributed wavelet transform framework; and (b) simply entropy coding quantized sample differences. The input source data are 12-bit integers produced as quantized version of a second order AR process with poles at $0.99e^{\pm j\pi/64}$. The gap between the curves represents the benefit of the combined distributed DWT and entropy coding approach compared to the simpler alternative.

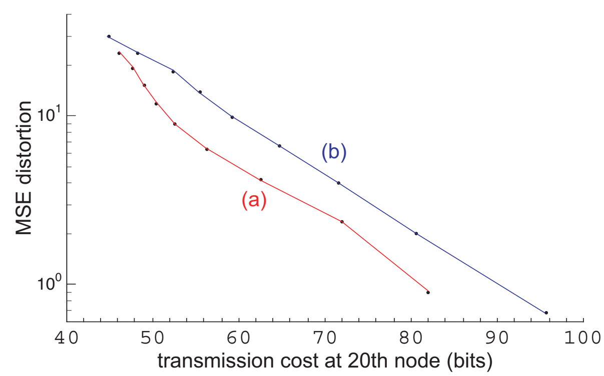

Fig. 6. Rate distortion performance comparing transmission cost at the 20th node in a chain **(a)** using the proposed entropy coding approach in a distributed wavelet transform framework, and **(b)** simply entropy coding the quantized sample differences.

4. BIT ALLOCATION

We have been investigating new bit allocation techniques for our distributed wavelet transforms. Standard bit allocation techniques for wavelet transforms achieve optimal bit allocation via subband coding. Given a particular filter bank (FB) structure with M synthesis FB outputs, the variance of each output is computed and used (possibly in conjunction with weighting factors as for biorthogonal FBs [28]) to allocate bits to each subband using techniques developed in [17]. Note that if data is sampled at regularly spaced intervals, and if the information being gathered can be modeled by a spatially stationary process, then the information in each subband (e.g., all the low-pass wavelet coefficients) can also be modeled as a stationary process, for which these standard bit allocation techniques are well suited.

However, the placement of nodes in wireless sensor networks is random and irregular in general. Existing techniques assign the same number of bits to all coefficients corresponding to a subband and cannot exploit the irregular sampling structure inherent in wireless sensor networks. This motivates the need for a new bit allocation technique that can adapt itself to any 2D deployment of nodes. Our goal here is to develop a better generic bit allocation technique for lifting transforms on irregularly placed nodes.

We propose a simple and intuitive solution to locally optimize bit allocation. Rather than estimate the variance of each of the M subbands produced by our filter, we can instead estimate the variance (over time) of the wavelet coefficient corresponding to each node. In this case we would have N variance values for N nodes rather than $M < N$ variance values for each subband. This localizes the bit allocation and should lead to gains in coding efficiency. We can achieve further improvements in bit allocation if we use locally adaptive filtering [5] in conjunction with the locally adaptive bit allocation already discussed. Computing the variances in this way leads to a bit allocation technique that is locally adaptive to any 2D wavelet transform and outperforms existing global allocation techniques using subband coding. Figure 7 shows an irregularly sampled 1D signal of length 500 with 50 unique sample points chosen randomly from 1 to 500. The Signal to Noise Ratio (SNR) versus Energy Consumption (denoted as Cost) plot on the right shows that Locally Adaptive Bit Allocation (LABA) and Locally Adaptive Bit Allocation with Locally Adaptive Filtering (LABA/LAF) are both superior to a global subband allocation method.

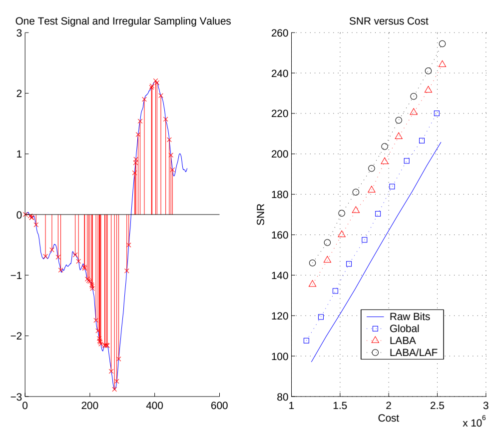

Fig. 7. Left shows a random irregularly sampled signal and right shows the SNR versus Cost plot for the various bit allocation techniques.

In a similar spirit, we have also developed a bit allocation technique that is a slight variant on the technique developed in [17]. Instead of imposing a rate constraint, we constrain the total energy consumption. Via standard Lagrange multiplier techniques, we are able to derive a bit allocation scheme that is very similar in form to the solution derived in [17] with the exception that the energy consumption (a linear function of sum of squared distances along each hop to the sink) per node is also accounted for implicitly and is fixed. A comparison between the LABA scheme and this fixed cost for minimum MSE scheme will be done for a unilateral 2D transform in the following section, see Figure 9.

5. NEW DISTRIBUTED TRANSFORMS

We now consider improvements to our existing system by introducing novel transform techniques. The existing system essentially uses a series of independent 1D wavelet transforms, with one transform performed along each route in the network. Because routes merge on their way to the sink, a simple differential encoding is applied where paths overlap [5] so that overall data transmission is reduced. However, this approach for "merging" two 1D transforms is suboptimal in that it is essentially non-critically sampled, i.e., additional information has to be sent for each path merge, whereas a critically sampled transform would require sending only as many coefficients as there are nodes in the network. In addition, this does not fully exploit data correlation across multiple paths (particularly at merge points). The main advantages of this approach are simplicity, and the fact that the transform is inherently tied to an efficient routing structure and is performed in a unidirectional manner (no backwards data transmissions).

Other distributed 2D lifting transforms done in [15,29,30] can achieve more data de-correlation and hence more efficiently coded data since more than two neighboring nodes per node participate in the transform as is the case in our existing system. However this leads to situations where at some nodes data may need to be sent away from the sink, resulting in backward data transmission. Such transforms may also require nodes to transmit data to other nodes that are not very close spatially, particularly at coarser levels of decomposition. These transforms may improve coding efficiency but do so at the expense of potentially much higher energy consumption than our existing method. Clearly a balance between increased coding efficiency for increased cost need be met.

Our goal in this work is to design a distributed 2D transform that can be computed in a unilateral fashion along an efficient routing structure, thereby achieving our desired balance. The transform should be critically sampled to avoid the overhead in [5, 7], while avoiding the higher cost associated with distributed 2D transforms. A simple way to do this is to perform the same 1D unilateral transform wherever paths do not overlap and to perform a 2D transform wherever paths merge.

Consider the random sensor deployment with a shortest path routing tree shown below in the left side of Figure 8. In order to apply a lifting transform to such a network, we must first split the nodes into even and odd nodes at each level of decomposition. Note that the sets of even and odd nodes need not be equal for a general lifting transform. Various methods of splitting the nodes exist including use of Delaunay triangulations as in [12,30]. One drawback with these splitting techniques is that they are not well matched to any particular routing tree. Some data exchange between nodes that are not connected in a given routing tree may also be necessary, in some cases imposing large cost if nodes are far apart.

One natural way to ensure that the transform is performed along a routing tree is to split the nodes up according to their depth in the tree with respect to the tree's root node as in the right side of Figure 8. In our case, the root node is the network sink. Using this notion of splitting, we could assign nodes to be even (update) or odd (predict) in one of two ways. Either nodes of odd depth are odd nodes and nodes of even depth are even nodes, or vice versa. Suppose we use the first splitting method. We then achieve a critically sampled transform that is perfectly aligned with the routing. We call node A a child of node B and node B a parent of node A if node A forwards its data to node B. We also call node A a neighbor of node B.

A simple version of this transform works as follows at merge points. For an even merge node we multiply its neighboring detail coefficients by $\frac{1}{2}$, sum, then add this sum to the merge node's data. For an odd merge node we multiply the raw data of its neighboring

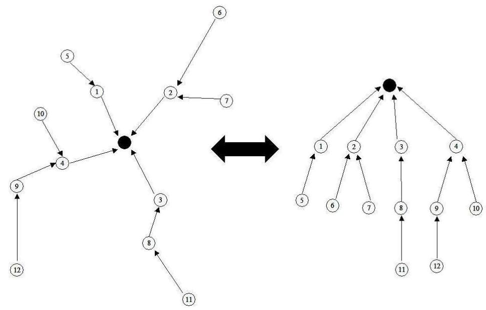

Fig. 8. Shows the equivalent routing structures, where right figure indicates how to decimate the transform. Nodes of odd depth are odd (1-4, 11, 12) and nodes of even depth are even (5-10).

nodes by $-\frac{1}{4}$, sum, then add this sum to the merge node's data. We can easily extend the partial coefficient approach developed in [7] by computing and transmitting separate partial coefficients for each coefficient at the merge node using the exact same computations as in [7]. The parent of the merge node will then complete the computation of the separate coefficients using computations detailed in [7]. A simple performance comparison is done as shown in Figure 9. The data in this case is highly correlated with 100 nodes randomly placed on a 600x600 grid. The unilateral transform is clearly superior to our existing system, and even more so when we employ the bit allocation scheme that achieves minimum MSE for a fixed cost.

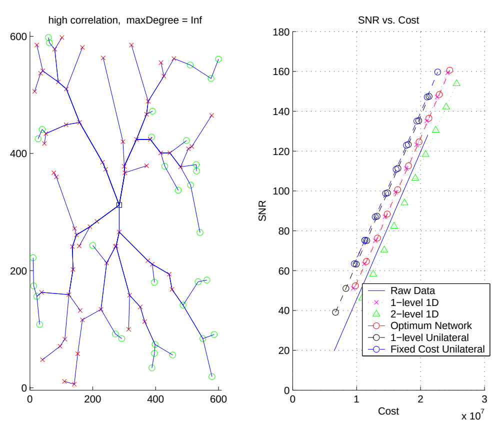

Fig. 9. Left shows routing topology and optimum transform for each node in the network. Right shows SNR versus Cost plots for existing methods compared with new Unilateral 2D method.

6. COMPRESSED SENSING

A common characteristic of all the transforms described in the previous section is that they transmit at least one wavelet coefficient per node for each sampling time. Thus the power consumption in the system is directly proportional to the number of nodes. We are also considering techniques that exploit known characteristics of the data being measured to make it possible to encode less than one coefficient per node on average.

To achieve this goal we are investigating the application of compressed sensing (CS) techniques. CS is a promising method that can reconstruct a K-sparse signal, x, with the size of N from M measurements of the signal [3, 13]. The measurements, $y \in R^M$, are obtained via the linear matrix-vector multiplication $y = \Phi x$, with $K \ll M \ll N$. The measurement matrix (Φ) which represents how the measurements are formed from samples is a $M \times N$ matrix whose elements can be random coefficients such as Bernoulli or Gaussian random variables.

To apply CS to wireless sensor network, we consider energy consumption and routing, which previously has not been explicitly taken into account in the context of CS. In most previous work in the field of CS, the measurements are linear combinations of every sample of input signal, i.e., Φ is a full matrix. This approach can not be directly applied to wireless sensor network due to its inherently high energy consumption. Based on the assumption that energy is dissipated only during data transmission among sensors, we need to design an algorithm that efficiently collects M measurements then transmits them to the sink.

For energy efficiency, we consider a sparse measurement matrix which contains a few non-zero random coefficients as its elements for energy-efficiency. Previous work [24] has proposed a new CS scheme using sparse measurement matrix similar to the sparse parity check matrix in LDPC instead of full measurement matrix. In that work, the sparse measurement matrix showed comparable performance to the full matrix in terms of the number of measurements and the reconstruction quality [24]. Taking this idea as a starting point, we are seeking to design an algorithm to construct a sparse matrix which minimizes the energy consumption due to data transmission as well.

The algorithm has to meet three requirements. First, each row of Φ contains exactly L non-zero entries (1 or -1). This means that every measurement consists of the information from L sensors. L is chosen based on the properties of the signal that we are trying to transmit (such as sparsity). Second, the non-zero entries are uniformly distributed over columns; in other words, every sensor has an equal chance to provide the information about its measured data for measurements. Lastly, given inter-sensor distances, the solution of the algorithm achieves minimum energy consumption. For simplification, we assume that the energy consumption can be evaluated as the product of the number of bits to be transmitted and the squared distance to the sink.

7. ERASURE-CORRECTING CODES

A major challenge in networking the low-power low-capability radios of the sensor nodes is that many communication links will be highly unreliable and lossy, showing asymmetry and large temporal fluctuations, due to multipath fading effects and individual hardware variance. We are investigating several approaches to improving the reliability of network communications, including routing algorithms, network coding, and channel coding on individual links. Our work to date has included an investigation of rateless erasure-correcting codes suitable for application to node-to-node links subject to large fluctuations in link availability.

Our link model is that transmissions between nodes are subject to erasures with unpredictable probability p_ϵ. Without coding, erased transmissions are retransmitted to the next node until the message gets through. This is efficient only if the sending node knows which transmissions (or portions thereof) are erased. Complicated erasure patterns require the use of complicated reverse-channel link protocols to inform the sending node which transmissions need retransmission. Alternatively, without knowledge of when erasures occur, the sending node can transmit pieces of data at random until the message gets through. This requires only a simple acknowledgment protocol over the reverse channel, but in this case the sending node needlessly retransmits non-erased data as well as erased data. Either alternative can be very costly in terms of the overall energy consumption of the two nodes.

Fixed-rate forward erasure-correcting codes offer a traditional solution to providing reliable communication with minimal energy consumption. However, when the link's erasure statistics are unpredictable, fixed-rate codes can spend too much overhead if the channel is better than expected, or fail altogether if the channel is worse than expected. A better solution is an erasure-correcting code whose rate adapts to the actual link statistics.

A *rateless* forward erasure-correcting code is one that communicates a fixed number k of information symbols by transmitting a variable number n of coded symbols. Here a "symbol" can refer to an individual data bit or to an entire data packet. Examples of rateless codes are Luby's *LT codes* [21], Shokrollahi's *Raptor codes* [26], Studholme and Blake's *windowed erasure codes* [27], and *random rateless codes* [27]. With a rateless code, the actual number of transmitted symbols n can vary according to the quality of the channel. Only a simple acknowledgment protocol is required over the reverse channel to tell the sending node when the k information symbols are successfully decoded.

The inventors of LT and Raptor codes measure the overhead $k^+ - k$ of a rateless code on a pure erasure channel in terms of the number of coded symbols k^+ (out of n) that are successfully received (not erased). LT codes and Raptor codes are asymptotically optimal for channels with an arbitrary and unknown erasure rate. Such codes of unbounded input block size are capacity-achieving in that they can achieve arbitrarily small frame erasure probability with arbitrarily small percentage average overheads $\epsilon = \left(E\{k^+\} - k\right)/k$ as $k \to \infty$. Large LT codes and Raptor codes with practical finite input blocks can achieve extremely small frame erasure probabilities with maximum average overheads ϵ of a few percent. Smaller Raptor codes can achieve moderately low frame erasure rates with maximum average overheads of a few percent.

Random rateless codes can achieve zero frame erasure probability on the erasure channel with fantastically small average overhead $E\{k^+\} - k < 1.61$ symbols, even with small finite input blocks, if they are decoded optimally using a "full-rank decoder" of much higher complexity than the decoder for LT or Raptor codes. Windowed erasure codes mimic random rateless codes but use a much sparser density of graph connections. They also achieve very small average overhead $E\{k^+\} - k = 2$ to 3 bits, and require "full-rank" decoding. As with random rateless codes, their average overhead is bounded by a small fixed number of bits independent of the code block size. Decoding complexity of windowed erasure codes is intermediate to that of LT or Raptor codes and fully random rateless codes due to the built-in sparseness of the code graph.

For our sensor web application, the relevant question is: What is the best way to construct a rateless forward erasure-correcting code that minimizes overhead – even for small input blocks – while maintaining reasonable decoding complexity? Uncoded resends are trivial to decode but are very inefficient, requiring on average $E\{k^+\} \sim k \ln k$ non-erased transmissions to decode k symbols successfully. LT and Raptor codes are reasonably easy to decode, using $O(k \ln k)$ or $O(k)$ operations, and are much more efficient, especially for large block sizes, requiring $E\{k^+\}$ to exceed k by only a few percent. Windowed erasure codes are even more efficient, requiring $E\{k^+\}$ to exceed k by only 2 to 3 symbols, even for small code blocks, but their decoding complexity is $O(k^{3/2})$. Random rateless codes are the most efficient, with $E\{k^+\}$ exceeding k by less than 1.61 symbols, but their decoding complexity is $O(k^3)$. The best choices for our application are windowed erasure codes if the nodes' computational costs are trivially small relative to their communication costs, or LT or Raptor codes if the computation and communication costs are more comparable.

8. IMPLEMENTATION

The hardware platform being used for in-lab implementation/testing is the TMote Sky wireless sensor module [1]. It uses the 2.4 GHz frequency for radio communication and has a data rate of 250 Kbps. The available RAM is 10 KB, program space is 48 KB and external flash memory is 1MB. The output power can be varied between -25 dB and 0 dB. Onboard sensors provide photo, temperature and humidity measurements. The detail schematic of the TMote Sky device is available online [2].

Our current in-lab implementation is based on the following assumptions: i) 1-level 5/3 wavelet compression using lifting, ii) Linear topology (fixed routing), iii) clear separation of epochs (a single data value collected from each node in one epoch), iv) arbitrary (configurable) bit allocation at each node, and v) nodes are controlled from a command center (laptop). We have implemented the distributed 1-level 5/3 wavelet using the the partial coefficient equations proposed, with uniform quantization. However, this is only for a single data sample per node in a linear topology with known bit allocation. For concretely demonstrating compression gains, additional components will be required as follows.

First, we will need to develop a *training phase* where nodes collect data samples/coefficients and route it to base station. The base station then has to calculate the relative variance in the node measurements and the optimal bit allocation based on the budget. This allocation has to be propagated back to the nodes. The procedure has to be repeated at certain intervals based on knowledge of the phenomena/temporal variations expected. Alternately, nodes can collect a specific number of samples/coefficients, calculate range and variance information and send only this information to the base station which will then propagate the bit-allocation back to nodes. This will lower the communication cost for training.

Second, support of *multiple levels of wavelet decomposition* will be needed, i.e., from the training data, the base station also needs to determine if, and at which nodes, multiple levels of decomposition will provide compression gains and propagate the same back to the nodes.

Third, *packetization/Stuffing strategies* need to be developed to group quantized coefficients into packets. Clearly, transmitting packets with individual coefficients is inefficient. For obtaining compression gains, it is required to pack the full/partial coefficients into packets of known byte-lengths. The sensor measurements at each node have to be accumulated and stored. When a packet containing several partial coefficients is received from an upstream node, the sensed values at the current node have to be combined with the matching coefficients for the current lifting step. We are investigating whether there is an optimal packet length and strategies to operate with *update* and *history*, as described in Section 3. Decoding previously transmitted data to predict newly acquired one, and provide more efficient transmission of *update* information, will incrase computation/delay at each node. Instead *history* information only needs to be relayed towards the base station and creates minimal delay or computation overhead.

Last, we need to consider *storage* issues when determining where and how to store sensor measurements. There are two main options. We could use RAM if a limited number of samples has to be stored, with the advantage of fast and easy access. Alternatively Flash storage may be considered if a large number of samples needs to be stored. Because of the delay implications this would have, our various algorithms should be evaluated not only in terms of computation and delay, but also in terms of their storage requirements.

8.1. Collaboration plans

We are pursuing several avenues to define specific science environments for which to customize our techniques and on which to deploy simple test systems, if possible. In considering what is feasible we are taking into consideration the capabilities of the motes (our target sensor development and testing platform.) We are in particular focusing on what can be done given the measurement sensor types, communication ranges, etc. The JPL investigators have started to develop a plan to select specific NASA applications that could be suitable to demonstrate our techniques. Further, we have identified a target environment for demonstrating the effectiveness of our compression techniques. AIMS (Australian Institute of Marine Sciences) is deploying WSNs to monitor growth, development and health of the corals at the Great Barrier Reef. Our aim is to set up a long-standing (greater than 1 month) medium size (50-100 motes) WSN test bed in conjunction with AIMS. The plan is to implement and test joint routing and compression algorithms for data collection from the test bed, in addition to non-trivial tree construction and sleep scheduling algorithms developed by ANRG.

9. CONCLUSIONS

In this paper we have provided an overview of a collaborative project that is designing new approaches for gathering, compression and representation of spatially correlated data in a sensor network. This project spans a range of issues, from signal representation and compression optimized for 2D irregularly sampled measurements, to the design of efficient erasure codes to ensure reliable operation. We are working on a testbed system to validate our designs.

10. REFERENCES

[1] *http://www.moteiv.com/products/docs/tmote-sky-brochure.pdf*.

[2] *http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf*.

[3] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. on Information Theory*, 52(2):489 – 509, February 2006.

[4] C. Chong and S. P. Kumar. Sensor networks: Evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, August 2003.

[5] A. Ciancio. *Distributed Wavelet Compression Algorithms for Wireless Sensor Networks*. PhD thesis, University of Southern California, 2006.

[6] A. Ciancio and A. Ortega. A distributed wavelet compression algorithm for wireless sensor networks using lifting. In *Proceedings of the 2004 International Conference on Acoustics, Speech and Signal Processing - ICASSP04*, Montreal, Canada, May 2004.

[7] A. Ciancio and A. Ortega. A distributed wavelet compression algorithm for wireless multihop sensor networks using lifting. In *Proceedings of the 2005 International Conference on Acoustics, Speech and Signal Processing - ICASSP05*, Philadelphia, USA, March 2005.

[8] Alexandre Ciancio and Antonio Ortega. A dynamic programming approach to distortion-energy optimization for distributed wavelet compression with applications to data gathering inwireless sensor networks. In *ICASSP'06: Proceedings of the 2006 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2006.

[9] Alexandre Ciancio, Sundeep Pattem, Antonio Ortega, and Bhaskar Krishnamachari. Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm. In *IPSN '06: Proceedings of the Fifth International Conference on Information Processing in Sensor Networks*, pages 309–316, New York, NY, USA, 2006. ACM Press.

[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2001.

[11] R. Cristescu, B. Beferull-Lozano, and M. Vetterli. Networked Slepian-Wolf: Theory and algorithms. *1st European Workshop on Sensor Networks EWSN 2004*, 2004. Berlin, Germany.

[12] I. Daubechies, I. Guskov, P. Schroder, and W. Sweldens. Wavelets on irregular point sets. *Phil. Trans. R. Soc. Lond. A*, 357(1760):2397–2413, September 1999.

[13] D. Donoho. Compressed sensing. *IEEE Trans. on Information Theory*, 52(4):1289 – 1306, April 2006.

[14] R. Gallager and D. C. Van Voorhis. Optimal source codes for geometrically distributed integer alphabets. *IEEE Transactions on Information Theory*, IT-21:228–230, March 1975.

[15] D. Ganesan, D. Estrin, and J. Heidemann. Dimensions: Why do we need a new data handling architecture for sensor networks? *ACM SIGCOMM Comput. Commun. Rev.*, January 2003.

[16] M. Gastpar, P. Dragotti, and M. Vetterli. The distributed Karhunen-Loève transform. In *Proceedings of the 2002 International Workshop on Multimedia Signal Processing*, St. Thomas, US Virgin Islands, December 2002.

[17] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

[18] A. Goel and D. Estrin. Simultaneous optimization for concave costs: Single sink aggregation or single source buy-at-bulk. In *SODA*, pages 499–505, 2003.

[19] S. W. Golomb. Run-length encoding. *IEEE Transactions on Information Theory*, IT-12(3):399–401, July 1966.

[20] A. Kiely and M. Klimesh. Generalized Golomb codes and adaptive coding of wavelet-transformed image subbands. *JPL IPN Progress Report*, 42-154:1–14, June 2003.

[21] M. Luby. LT codes. In *Proc. 43d Annual IEEE Symp. Foundations of Computer Science (FOCS)*, pages 271–280, Nov 2002.

[22] S. Pattem, B. Krishnamachari, and R. Govindan. The impact of spatial correlation on routing with compression in wireless sensor networks. In *Proceedings of the Third International Symposium on Information Processing in Sensor Networks*, April 2004.

[23] S. S. Pradhan, J. Kusuma, and K. Ramchandran. Distributed compression in a dense microsensor network. *IEEE Signal Processing Magazine*, pages 51–60, March 2002.

[24] S. Sarvotham, D. Baron, and R. G. Baraniuk. Sudocodes - fast measurement and reconstruction of sparse signals. 2006.

[25] S. D. Servetto. Sensing Lena - massively distributed compression of sensor images. *ICIP - International Conference on Image Compression*, September 2003.

[26] A. Shokrollahi. Raptor codes. *IEEE Transactions on Information Theory*, IT-52:2551–2567, June 2006.

[27] C. Studholme and I. Blake. Windowed erasure codes. In *2006 IEEE International Symposium on Information Theory*, pages 509–513, July 2006.

[28] B. Usevitch. Optimal bit allocation for biorthogonal wavelet coding. In *Data Compression Conference*, page 387, Los Alamitos, CA, USA, 1996. IEEE Computer Society.

[29] R. Wagner, R. Baraniuk, S. Du, D.B. Johnson, and A. Cohen. An architecture for distributed wavelet analysis and processing in sensor networks. In *IPSN '06: Proceedings of the Fifth International Conference on Information Processing in Sensor Networks*, pages 243–250, New York, NY, USA, 2006. ACM Press.

[30] R. Wagner, Hyeokho Choi, R. Baraniuk, and V. Delouille. Distributed wavelet transform for irregular sensor network grids. July 2005.

[31] Marcelo J. Weinberger, Gadiel Seroussi, and Guillermo Sapiro. LOCO-I: A low complexity, context-based, lossless image compression algorithm. In *Data Compression Conference*, pages 140–149, 1996.